REMARKS

Entry of the preceding amendments to claims 13, 23, and 24 is respectfully requested, as is a reconsideration of claims 13 through 24 in view of the comments to follow.

The present amendment is respectfully submitted in response to the Office Action of November 27, 2007. In that action, claims 13 through 24 were rejected on the basis of the prior art.

More specifically, referring to page 2 of the action, claims 13 through 24 were rejected under 35 U.S.C. § 103(a) as being unpatentable for obviousness over Saboff (U.S. Patent No. 6,154,878) in view of Segal ("On-the-fly Program Modification: Systems for Dynamic Updating") and further in view of Tang et al. (U.S. Patent No. 7,209,449).

In order to overcome these rejections, claims 13, 23, and 24 have been amended as set forth above. Support for the amendments may be found in the specification, for example, at page 11, lines 18 through 29.

As claimed in the claims as amended above, the present invention comprises "an execution instruction section formed by a computer program", which performs the following three operations:

1. to maintain a data-pointer for specifying at least one of the plurality of data stored in said data storage section, the data-pointer being maintained as a value of a data-pointer variable defined in the computer program;

2. to await occurrence of the corresponding event; and

3. to call the execution section specified by the pointer, when occurrence of the corresponding event is detected, giving the data-pointer variable to the execution section as an

argument, thereby causing the execution section to execute computation by use of said at least one of the plurality of data as specified by the value of the data-pointer variable.

In short, the execution instruction section in the claimed invention maintains a value of a data-pointer variable defined in the computer program, and calls an execution section to which the data-pointer variable is given as an argument such that the execution section executes computation by use of the data specified by the value of the data-pointer variable.

The data specified by the data-pointer as above has been stored in the data storage section and kept after the pointer for specifying the execution section is changed, in the claimed invention.

Therefore, after the first execution section is replaced with the second execution section, the execution instruction section in the claimed invention can call the section execution section, giving it the same data-pointer variable that has been given to the first execution section, the value of which is maintained by the execution instruction section.

By this mechanism, the claimed invention enables the second execution section to use the same data that has been used by the first execution section, and thus enables the execution section (or event handler) to be updated while keeping a state of communication or socket in the router.

In contrast, Saboff discloses the system comprising an application 30 which is linked with an interface library 32, an implementation library 34 which is called from the application via the interface library, and management services 36 (see Fig. 2). In this system, "the implementation library must manage its data such that the state of the library, i.e., the values of its local variable, can be restored when an update to the implementation library occurs" (col. 3, lines 6-9), and for this purpose "the memory management services" are provided, which "allow

the implementation library and interface libraries to manage memory so as to preserve the state of the implementation library from the old version to the new version" (col. 3, lines 28-31).

That is, Saboff's mechanism for the new implementation library to use the same data that has been used by the old implementation library after updating is actualized by the implementation library itself or the newly-provided memory management services (see also col. 14, lines 55-60, "Memory needs to be managed so that the state of the implementation library can be recovered after the implementation library has been changed. This management could be done by services within the implementation library, but is optimally done by the use of external memory management services," and col. 15, lines 11-14, "The memory management services must be used by the implementation library for the allocation of all memory to ensure that the state of the library can be recovered after swapping a library"). Therefore, in Saboff's system, "no changes to existing applications are needed for them to run successfully in a system that implements replaceable libraries" (col. 5, lines 34-36).

The Saboff's teaching is the opposite to the mechanism of the claimed invention. In Saboff, there is no special requirement on the application which calls the implementation library, whereas the implementation library which is to be updated must be specially arranged or the external memory management services must be newly provided.

On the contrary, according to the claimed invention, the execution instruction section which calls the execution section must be arranged to perform the three operations as recited above, whereas the execution section which is to be updated can be the conventional one and there is no need of any special management services.

In addition, Saboff's system must use a special memory management scheme by either the implementation library or the memory management services, as shown in Fig. 8, in

which each implementation library contains a module header block 150 including an enduring head pointer 152, which indicates an enduring data header block 160 containing an allocation ID 161. This memory management scheme must be referred to, every time the implementation library allocates memory, and thus would slow down the execution of computer software.

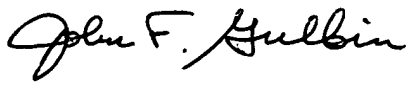
The claimed invention can eliminate such requirement of the special memory management scheme, due to the new arrangement of the execution instruction section as explained above, and thus enables an internet packet router to be partially updated in its software, which keeps executing computation on routing information, without obstructing the continuous computation.

Consequently, Saboff does not teach or suggest "the execution instruction section" which performs the three operations as recited in the amended independent claims of the Applicant's invention. Neither does Tang et al. or Segal. Therefore, the claimed invention would not be obvious to the ordinarily skilled in the art at the time of the invention in view of the teachings of the cited references considered singly or in any combination with one another.

Accordingly, claims 13, 23, and 24 are respectfully submitted to be patentable over the combined teachings of Saboff, Segal, and Tang et al. Moreover, claims 14 through 22, which all depend from claim 13, are respectfully submitted to be patentable over the combined teachings of these cited references as further limiting the subject matter of an allowable claim 13.

In view of the above, an early allowance of claims 13 through 24 is requested and earnestly sought.

Respectfully submitted,



John F. Gulbin
Registration No. 33,180

Day Pitney LLP
7 Times Square
New York, New York 10036-7311
212-297-5800